# Conceitos iniciais

## Video

Sequência de imagens

## Áudio

Ondas sonoras convertidas em sinais analógicos, armazenados em sinais digitais. Ao reproduzir, os sinais digitais são reconvertidos em analógicos para então virarem ondas sonoras novamente.

## Codec

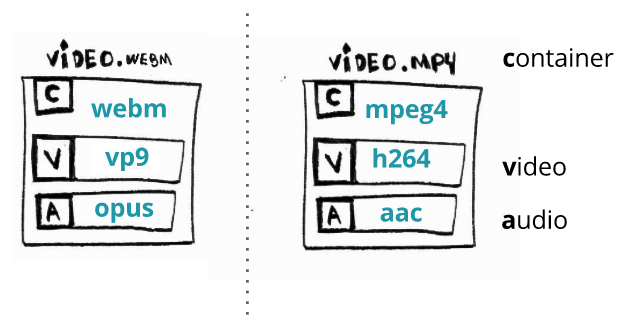
Compressão e descompressão de dados de áudio/vídeo.

Exemplos: h264, avi, webm…

## Container

Cápsula de união de áudio e vídeo.

Alguns containers possuem o mesmo nome do CODEC.



## Protocolos de transmissão de vídeo

Nesse caso estamos falando de como o vídeo é enviado de um ponto a outro pela internet.

Exemplos: RTSP, webRTC, DASH, RTMP.

# Onde está o maior consumo de processamento?

No CODEC! Compressão, descompressão e alterar o codec de um vídeo é o trabalho mais pesado! Evite sempre que possível!

Você consegue alterar o container sem alterar o formato da compressão (Codec) do vídeo (na maioria das vezes)!

# Como trabalhar com vídeo

Usando frameworks! Não adianta… mexer com bits dos vídeos não vale a pena!

Vou exemplificar abaixo com FFMPEG, porém outros frameworks funcionam de formas parecidas (GStreamer por exemplo).

Ffmpeg funciona usando seu executável como o início para chamadas por linha de comando. Baixe seu ffmpeg.exe e a partir da pasta comece a fazer as chamadas ffmpeg.

## Usabilidade básica

ffmpeg.exe -i input.avi output.mp4

nesse primeiro exemplo inserimos um arquivo chamado input.avi com container avi e transformamos em output.mp4 com container mp4. O próprio ffmpeg consegue descobrir (na maioria das vezes) o codec de entrada e de saída.

## Manipulando codec

ffmpeg.exe -i input.avi -c copy output.mp4

Agora estamos falando para o framework copiar o codec de entrada para a saída. Nesse caso temos uma conversão mais rápida sem alterar o codec. A flag -c significa CODEC

ffmpeg.exe -i input.avi -c:v libx264 -c:a 137 output.mp4

Nesse caso estamos falando para converter usando a biblioteca libx264 para o CODEC h264 (vídeo) e o CODEC 137 no áudio. -c:v significa codec de vídeo. -c:a significa codec de áudio.

## Transmitindo para a internet

Podemos fazer com que o ffmpeg faça transmissão direta para um servidor de vídeo (OpenVidu, SRS, Node-Media-Server). Para isso, devemos especificar diretamente a url de recepção de vídeo do servidor. Da mesma forma podemos colocar a url de entrada diretamente.

ffmpeg.exe -i rtsp://admin:12345@192.168.1.7:8554/live -c copy -f flv rtmp://172.25.150.10/live/livestream

A flag -f indica ao ffmpeg que desejo forçar a saída do container como um flv.

## Outras flags úteis a serem consideradas

Veja a documentação para mais detalhes

-avoid\_negative\_ts make\_zero

-fflags +genpts+discardcorrupt

-hide\_banner

Existem também formas de reduzir resolução, controlar taxa de bitrate, salvar fotos ao decorrer do vídeo, etc… google it.

## Como trabalhar em C#

Apesar de existir frameworks que fazem chamadas no FFMPEG, simplesmente iniciar uma chamada de linha de comando é uma das formas mais fáceis de trabalhar. FFMPEG fecha quando dá um erro, então dá para usar para validar o status do processo.

# Servidores de vídeo

[Centralizador de video](https://docs.google.com/spreadsheets/d/1I6v8Pe_1TBPoYoeFBvLAqP3Ph2ualbgyOjyam5vgN_s/edit#gid=0)

